2

NPS-AS-92-025

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AN INITIAL MODEL OF REQUIREMENTS TRACEABILITY AN EMPIRICAL STUDY

Balasubramaniam Ramesh
Ann G. Abbott
Mona R. Busch
Michael Edwards

September 1992

Approved for public release; distribution is unlimited.

Prepared for: Naval Surface Warfare Center Dahlgren Division
Silver Spring, MD 20903-5000

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R.W. West, Jr.
Superintendent

H. Shull
Provost

This report was prepared for and funded by the Naval Surface Warfare Center, Dahlgren Division, Silver Spring, Maryland.

This report was prepared by:

Balasubramaniam Ramesh
Department of Administrative Sciences

Reviewed by:

for

David R. Whipple, Chairman
Department of Administrative Sciences

Released by:

Paul J. Marto, Dean of Research

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	22 September 1992	Technical Report, September 1992

4. TITLE AND SUBTITLE

An Initial Model of Requirements Traceability: An Empirical Study

5. FUNDING NUMBERS

N6092192WRW0067

6. AUTHOR(S)

Balasubramaniam Ramesh, Ann G. Abbott, Mona R. Busch and Michael Edwards

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

Administrative Sciences Department
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-AS-92-025

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Surface Warfare Center Dahlgren Division
Silver Spring, MD 20903-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A primary concern in the development of large-scale, real-time, complex, computer-intensive systems is ensuring that the system meets the specified requirements. Further, the requirements themselves evolve and undergo many changes during the development process. In such a context, it is essential to maintain traceability of requirements to various outputs to ensure that the system meets the current set of requirements.

An empirical study, utilizing focus group and protocol analysis techniques, was conducted with students from the Naval Postgraduate School in a simulated systems development environment. Based on the analysis of data, this studey highlights several major issues that need to be considered in the development of a model of traceability. An initial model of traceability as well as recommendations for future research to refine and validate the model are presented.

14. SUBJECT TERMS

15. NUMBER OF PAGES

71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

AN INITIAL MODEL OF REQUIREMENTS TRACEABILITY

An Empirical Study

Balasubramaniam Ramesh, Ann Grayson Abbott, Mona Rose Busch
Code AS/RA
Naval Postgradutae School
Monterey, CA 93943

and

Michael Edwards
Code U33
Naval Surface Warfare Center Dahlgren Division
10901 New Hampshire Avenue
Silver Spring, MD 20903-5000

ABSTRACT

A primary concern in the development of large-scale, real-time, complex, computer-intensive systems is ensuring that the system meets the specified requirements. Further, the requirements themselves evolve and undergo many changes during the development process. In such a context, it is essential to maintain traceability of requirements to various outputs to ensure that the system meets the current set of requirements.

An empirical study, utilizing focus group and protocol analysis techniques, was conducted with students from the Naval Postgraduate School in a simulated systems development environment. Based on the analysis of data, this study highlights several major issues that need to be considered in the development of a model of traceability. An initial model of traceability as well as recommendations for future research to refine and validate the model are presented.

TABLE OF CONTENTS

I. INTRODUCTION

A. GENERAL INFORMATION

A primary concern in the development of complex, large-scale, real-time, computer-intensive systems is ensuring that the design of the system meets the specified requirements. As part of the systems development and maintenance process, many decisions and tradeoffs are made that affect a variety of the components. Further, the requirements themselves also undergo many changes and evolve during the development process. In such a context, it is essential to maintain the traceability of requirements to various outputs produced during the system's design process, ensuring that the system meets the current set of requirements.

The term traceability, as used in this research, refers to a technique used to provide relationships between requirements, design and implementation of a system (Edwards, 1991). A comprehensive scheme for maintaining traceability, especially for complex, real-time systems, requires that all system components (not just software), created at various stages of the development process, be linked to the requirements. These components include hardware, software, humanware, manuals, policies, and procedures. In order to achieve this objective, it is essential that traceability be maintained through all phases of the systems development process, from the requirements as stated, or contracted, by the customer, through analysis, design, implementation, and testing to the final product.

Maintaining consistency between the requirements and the design is especially critical in situations where an organization relies upon outside contractors for developing systems. Having a systematic way of validating that each requirement is met by the design is important, not only to ensure that the system performs correctly, but also to determine whether contractual obligations have been met.

The need to provide traceability is recognized in most critical standards governing the development of systems for the U.S. Government. However, a clear definition of the types of information or relationships between various system components that are part of a traceability scheme is lacking. For instance, the DoD-STD-2167A specifies that

> the contractor shall document the traceability of the requirements allocated from the system specification to each Computer Software Configuration Item (CSCI), its Computer Software Units (CSUs) and from the CSU level to the Software Requirements Specification (SRSs) and Interface Requirements Specifications (Walters, 1991),(DoD, 1988).

An elaboration of this requirement states that

> the Software Design Document describes the allocation of requirements from a CSCI to its CSCs and CSUs (Walters, 1991)

It should be noted that even this elaboration is not specific about the nature of traceability linkages to be maintained. Neither the standards that require traceability as a part of any systems development effort nor the current literature elaborate on the specific types of traceability linkages to be maintained. Though current tools provide mechanisms to represent various types of linkages between system components, the interpretation of the meanings of such linkages is left to the user. Unless the semantics of the traceability linkages are clearly specified, the existence of a link between a design element and a

requirement could denote one of several possibilities: the requirements have been completely allocated, some of the design elements partially satisfy a requirement, the fact that the design element satisfies a requirement can be formally verified etc.

Finally, the focus of much of existing research is on providing traceability at the level of software design, rather than at the level of system design.

B. OBJECTIVE OF THE RESEARCH

The goal of our research program is to develop a model of traceability at the level of systems design, relating requirements to all system components. Such a model should provide the semantics of the various traceability linkages or relationships between requirements and various system components. It should also provide mechanisms for reasoning with traceability information to support systems development and maintenance activities.

A first step towards the development of a comprehensive model is to understand the critical issues that relate to the capture and use of traceability information in systems development. A basic premise in the current research, whose results are reported in this document, is that development of a model of traceability could be geared toward the needs of various stakeholders at different stages of the systems development process.

A variety of stakeholders are involved in the systems development process, including project sponsors, project managers, analysts, designers, maintainers, testing personnel, and end users. The approach used in this research to identify their needs has been an empirical one. We have conducted a study to explore the traceability needs of

various stakeholders and to identify the critical issues that need to be addressed in the development of a comprehensive model of traceability. This study was conducted with graduate students of systems analysis and design in a simulated systems development environment. The results of this study are being used in designing a comprehensive study involving real stakeholders in large scale, complex, real-time systems development efforts.

Another objective of the current research is to evaluate different research tools for data collection and analysis to aid in the design of the comprehensive study. Two powerful research tools were employed in this research: protocol analysis to study problem solving behavior and focus group interviews for idea generation.

Given the above objectives, besides the development of an initial model of traceability, the following questions are addressed in this research:

- What are the critical issues that need to be addressed in the development of a model of traceability to support various stakeholders in systems development?
- What are appropriate methodologies that could be used in a comprehensive study on traceability in complex, large scale, real-time systems development environments?

C. SCOPE AND LIMITATIONS OF THE STUDY

The current study has employed novice systems designers in a simulated design setting. It should be noted that this research is designed to provide only insights into issues that need to be investigated further, rather than to provide conclusive results.

Another constraint was the lack of resources to comprehensively evaluate the current tools that support representing traceability information.

D. ORGANIZATION OF DOCUMENT

The document is organized as follows:

Chapter II provides background information on the general topic of traceability, a discussion of some of the current traceability tools available, and the uses of traceability in the DoD and U.S. Navy.

Chapter III describes focus groups and protocol analysis and their applications in this research. Each of the techniques, and why they were used in our research, is explained.

Chapter IV provides the analysis of the data collected, utilizing focus groups and protocol analysis techniques. It discusses the major findings and relates them to current literature.

Chapter V discusses design rationale as an example of traceability highlighting some of the major issues discussed in the previous chapter.

The final chapter provides an initial model of traceability and draws conclusions based on research data, and makes specific recommendations resulting from the research effort. This chapter concludes with recommended areas for additional research.

E. ACKNOWLEDGEMENTS

The authors would like to thank their sponsors, the Naval Surface Warfare Center Dahlgren Division, the Office of Naval Technology, especially Elizabeth Wald and Cmdr.

Gracie Thompson. The authors would also like to thank Phil Hwang and Steve Howell and all those who provided technical support in this research effort. Finally, thanks are due to the participants of the Systems Specification Synthesis Working Groups that helped refine some ideas investigated in this research.

II. BACKGROUND

A. WHAT IS TRACEABILITY?

A number of different definitions can be provided for traceability, depending on the context in which the term is used. Norman Schneidewind depicts traceability as a means for maintenance, focusing on the maintenance phase to discover sources of error. He defines traceability as "the ability to identify the technical information which pertains to a systems error which has been detected during the maintenance phase and thereby trace the error to the applicable design specifications and user requirements" (Schneidewind, 1982). Whereas Schneidewind's concern for traceability is at the software level, Greenspan and McGowan are concerned with the use of traceability to effect changes in the entire system at various levels. They offer a broad definition of traceability as being:

> a property of a system description technique that allows changes in one of the three system descriptions--requirements, design specifications, implementation--to be traced to the corresponding portions of the other descriptions. The correspondence should be maintained throughout the lifetime of the system (Greenspan and McGowan, 1978).

To achieve the abovementioned correspondence, Agusa, et al, postulate that two-way traceability is required. They label traceability as bi-directional by saying, "A requirements description is traceable if each portion of the description can be traced to an originating requirement in its predecessor, and to a successor description" (Agusa et al, 1984).

While all of the above definitions focus on change/maintenance, other aspects of traceability are not emphasized. Michael Edwards offers a more generic and inclusive definition of traceability as a technique used to "provide a relationship between the requirements, the design, and the final implementation of the system" (Edwards, 1991). This definition of traceability has been used in our current research.

B. TRACEABILITY TOOLS AND CURRENT EXPECTATIONS

The initial concern with traceability was that of providing document traceability. According to Horowitz and Williamson, "Document traceability defines the existence of relationships between two document components" (Horowitz and Williamson, 1986). Traceability within documents assures that the source of information is distinguishable.

There are a number of existing traceability tools developed by industry. Some salient characteristics of the major ones, including Automated Requirements Traceability System (ARTS) from Lockheed, Teamwork/RQT from Cadre Technologies Inc., Requirements Tracer (RT) from Teledyne Brown, and Requirements and Traceability Management (RTM) from GEM-Marconi Ltd. are discussed below.

One of the earliest systems to capture and use traceability data was ARTS, a bookkeeping program developed to manage the requirements of a large, error-prone aerospace system. ARTS operates on a data base, including systems requirements and their characteristics. It allows for automated tracking of requirements as they are partitioned and apportioned to lower level requirements. ARTS provides upward and

downward traceability and data base management and output operations on requirement-related attributes selected by the user. Like ARTS, other current tools often focus on the database management issues related to maintaining links between requirements and differing components of the system. The following are the major characteristics found in current traceability tools:

- Allocation of requirements to targets/design elements
- Parsing and grouping of functional requirements
- Traceability between documents
- Interface with CASE tools (e.g., Teamwork, Software Through Pictures)
- Capture functional hierarchies
- Keyword searches
- Assign attributes for requirements or traceability relationships
- Customized report generation
- Graphic Interfaces

The tools differ in the degree of support provided and offer only a subset of the above functionalities. Current traceability tools tend merely to provide mechanisms to represent relationships without providing a comprehensive model of traceability to aid the use in defining these relationships. Also, as they lack sophisticated mechanisms to reason with the traceability information captured, their usefulness is severely restricted.

C. TRACEABILITY IN THE DOD AND NAVY

As one of the world's major buyers of large-scale, computer-based systems, the DoD takes a surprisingly detailed approach to the dilemma of detailing systems requirements. DoD standards may even provide an instructive checklist for content.

In February 1988, the Defense Department specified its requirements for systems development in its Military Standard DoD-STD-2167A. This standard formalizes the tracing of requirements (in documents) from the original set entailed by the customer, to the contractor's written requirements specifications, and to the design, test procedures, and results. DoD-STD-2167A mandates that requirements be traceable through the entire system. However, the standard states only that traceability is required, not what information is to be maintained to achieve this.

The DoD currently delineates its requirements to contractors in documents that are developed by numerous specialists in a format that may be thousands of pages long. Having a precise method for ensuring that requirements are met by the design is vital. With declining defense dollars, systems must last longer, with potential for major changes during their lifecycle. Therefore, a key element included in a request for proposal must be traceability, guaranteeing that current set of requirements are met by the evolving system.

One of the foremost issues in developing an efficient and effective system involves the maintenance of consistency between requirements and design. Such consistency entails meeting the initial requirements and maintaining requirements, design, and implementation consistent throughout the entire system life-cycle.

The current method used by the Navy to specify requirements uses mostly a narrative, natural-language format with supporting diagrams and charts. Ambiguities are frequent as natural-language specifications are inexact. If specifications are formally stated and can be transformed into designs in a formal manner, traceability between requirements and designs is a by-product of the design process itself. However, most specifications are natural-language and therefore mechanisms are needed to capture traceability information explicitly.

In light of some recent systems malfunctions that produced catastrophic consequences (major telephone service shutdowns, for example), it is now commonly understood that changes to intricate systems can result in unforeseeable and disastrous effects to important national defense systems. These problems possibly could be avoided if correct traceability methods are used along with proper maintenance of systems.

D. SUMMARY

Acquiring a greater understanding of the concepts of traceability is essential. A major challenge in this research is the development of a model that represents and provides the semantics of various traceability linkages or relationships between requirements and systems components.

III. DATA COLLECTION METHODOLOGIES

A. INTRODUCTION

In order to better investigate the traceability relationships, we used a two-pronged approach to data collection: focus group interviews for idea generation and evaluation, and protocol analysis of problem solving behavior.
This chapter discusses these two techniques and the design of the study that employed these two methodologies. Details of the research setting, subjects as well as the reasons for the use of data collection techniques are provided.

B. FOCUS GROUPS

1. What Are Focus Groups?

Focus group interviewing is possibly the most consistent qualitative marketing technique used today. Marketers and the media have had much to say recently about focus groups. Many advertising and research agencies believe focus groups to be among the most valid of exploratory research tools for their purposes. Today, in many marketing research organizations, group interviews are nearly as common as the traditional survey questionnaire.

A focus group interview is a semi-structured exchange with a small group of people. It is not a rigidly constructed question-and-answer session, but neither is it a free dialogue among group members; the group has a clear agenda. In his book, Focus

Groups: A Practical Guide for Applied Research, Richard Kreuger states, "A focus group can be defined as a carefully planned discussion designed to obtain perceptions on a defined area of interest in a permissive, non-threatening environment. The discussion is relaxed, comfortable, and often enjoyable for participants as they share their ideas and perceptions. Group members influence each other by responding to ideas and comments in the discussion" (Krueger, 1988).

Focus groups were originally called focused interviews. They were first used in the 1930s by social scientists as an alternative to the technique of using an interviewer with closed-ended questions and one respondent; the idea was that multiple respondents could make comments on issues they believed to be important while interacting with one another. In the 1940s, focus groups were used in the evaluation of audience responses to radio programs, and the observation of the effectiveness of wartime propaganda efforts. In the 1990s, although much of what we know about the focus group technique has come from market research, all professions, from academia to diplomacy and politics, to the social science and business worlds, are adopting this eminently versatile method.

The focus group interview is a highly flexible tool and as such is extremely popular. Focus groups are appropriate for exploratory analysis when little is known about a topic; for generating ideas and research hypotheses; for determining how groups of individuals think about current issues; for producing information, uncovering potential problems, and encouraging creativity. Today, focus group interviewing is considered to be a valid scientific method.

The focus group technique was used by the Reagan administration in 1988 (an election year) to determine the character and extent of the knowledge/opinion gap between the American public and government officials, in regard to American-Soviet relations. The Reagan team asked two suburban Philadelphia focus groups of "average citizens" to examine the ways in which a future Soviet-American summit meeting could be believably presented to the American people while simultaneously garnering popular support. Based on focus group responses, the team chose for the trip the theme, "A brighter future and a safer world for all people." The Philadelphia groups also helped determine some of the events of the trip, and with whom President Reagan would meet.

Focus group interviewing today usually involves seven to 12 individuals who discuss a particular topic under the direction of a moderator, who promotes discussion and ensures that the group stays on the subject. Smaller groups may be dominated by one or two members, while larger groups are difficult to manage, and limit participation by all members. A typical session will last from one and one half to two and one half hours.

2. Uses, Pros, and Cons of Focus Groups

Focus groups may be used as a method for testing hypotheses, especially when the researcher has strong reasons to believe his hypotheses are correct. The focus group technique is not without its critics, who maintain that focus groups don't provide "hard" data and that group members may be atypical of a larger population. But even the critics acknowledge that focus groups are useful for exploratory research where little is known about a topic.

The more commonly lauded uses of focus groups include:

- Generating research hypotheses that can be submitted to further research and testing, using more quantitative approaches;
- stimulating new ideas and creative concepts;
- diagnosing the potential for problems with a new program, service, or product;
- generating impressions of products, programs, services, institutions, or other objects of interest; and
- learning how respondents talk about the phenomenon of interest.

Some advantages of focus groups include:

- They are quicker and less costly than individual interviews.
- Direct contact with respondents allows for probing and clarification; the respondent can use his own words to express himself.
- Through group interaction, members tend to influence and change each others' opinions, and this shift can be studied; information and insights are provided that would not be available without the group's interaction.
- Focus Groups have a dynamic effect, encouraging creativity.
- Results are believable and easy to understand.
- There is much research and theory related to focus groups.

Some disadvantages of focus groups include:

- The sample size is limited.
- Groups may vary widely in their enthusiasm levels and responses.
- Responses are not independent and may be biased by one or more participants.

- Summarization and interpretation of responses may be difficult.
- The moderator has less control in a group setting than in a one-on-one interview.
- The moderator may bias results.

In their book, Focus Groups: Theory and Practice, David Stewart and Prem Shamdasani state, "We should not overlook the cases in which focus groups alone may be a sufficient basis for decision making. One example in an applied research setting would be the identification of flaws or serious problems with a new product or program that would necessitate redesign" (Stewart and Shamdasani, 1990).

When little is known about a particular subject, there are few good alternatives to focus groups. Focus groups are quicker and less expensive than individual interviews; one must simply recognize the potential for obscuring individual responses.

3. Group Dynamics

It is the characteristics of group members in relation to one another, and not just individual differentiation, that determine group behavior and performance. Focus groups should be structured to facilitate the goals of the researcher, while avoiding manipulation of the final results.

A recurrent supposition regarding focus groups is that superior data are obtained when members are strangers. However, Stewart and Shamdasani state:

> Generally, focus group sessions are preceded by 'get-acquainted' and 'warm-up' sessions that usually provide participants ample opportunity to get to know one another. Thus, the issue of acquaintanceship appears to be a matter of degree in most focus groups, and its influence appears modest at best (Stewart and Shamdasani, 1990).

Another concern regarding focus groups is the members' backgrounds.

> In general, interaction is easier when individuals with similar socioeconomic backgrounds comprise the group. Similarity of abilities, level of intelligence, and knowledge tends to facilitate communication at the same wavelength. Similarly, in culturally and racially homogenous group situations, it may be easier to encourage member participation. This suggests that focus groups should be designated to maximize interaction by assuring similarity with respect to socioeconomic status (ibid, 1990).

A highly homogenous group may be able to move through many questions quickly, while a highly heterogenous group may belabor even a couple of questions. But a small degree of variation within group characteristics is often a helpful way to obtain the contrast and variation that spark lively discussions.

Krueger advises:

> The focus group technique works well when all participants are on an equal basis. Participants should be grouped with care. Participants should be placed with others at the same level or status in the organization. (Krueger, 1988).

4. The Moderator

When a moderator/interviewer has little experience or prior knowledge in a field, the focus group technique can be ideal. In his treatise, Focus Groups: A Qualitative Research, David L. Morgan states:

> When the researcher is relatively new to an area, or puts a priority on not repeating the received wisdom in a field, focus groups have much to offer. The fact that group interviews can produce useful data with relatively little direct input from the researcher may be a distinct advantage, especially in comparison to other interviewing techniques (Morgan, 1988).

A designated moderator/interviewer does away with much of the distraction associated with the group having to develop its own leadership. With respect to the

discussion, the moderator may be highly directive or very non-directive--letting the discussion flow naturally as long as it remains on the topic. It is quite common for an interviewer to start with some general questions, then focus on more specific issues as the group proceeds.

The amount of direction provided by the interviewer does influence the type and quality of the data obtained from the group. The amount of structure and direction by the moderator must be determined by the broader research agenda, including types of information sought, degree of detail the information requires, and the manner in which the information will be used.

Discussion of issues relevant to the needs of the researcher occur most readily when the moderator takes a more directive and structured approach. When this occurs, however, participants discuss what is important to the researcher, not necessarily what they consider significant.

5. The Interview Guide

In setting the agenda for a focus group, the moderator must choose from among research questions to create the interview guide. An alternative, available to a researcher conducting several focus groups, is the <u>rolling</u> interview guide. The interview guide developed for Group One is revised and used for Group Two. Based on Group Two results, the guide is revised for Group Three, and so on. This technique makes the best use of multiple focus groups, permitting information to be refined over time as more information is obtained about a subject.

6. Analyzing Focus Group Data

According to Stewart and Shamdasani:

> The most common purpose of a focus group interview is for an in-depth exploration of a topic about which little is known. For such exploratory research a simple descriptive narrative is quite appropriate. More detailed analyses simply are not necessary or efficient (Stewart and Shamdasani, 1990).

For analyzing the content of focus groups, the cut-and-paste method is immediate and cost-effective. Cut-and-paste is a useful technique, but often relies on the judgment of a single analyst. Usually it is preferable to have two or more analysts code the focus group results independently.

C. PROTOCOL ANALYSIS

1. Definition of Protocol Analysis

Vitalari and Dickson define protocol analysis as "the process of translating the chaotic collection of information, which is derived from the protocol, into more useful and meaningful representation" (Vitalari and Dickson, 1983). In a more general sense, protocol analysis can be thought of as the collection and analysis of verbal reports (called protocols) made by subjects while they perform a specific task. In most cases, protocol analysis is used to generate a mechanism for tracing a subject's thought process.

Ericsson and Simon distinguish between two different types of verbalization procedures--retrospective verbalization and concurrent verbalization. Retrospective verbalization refers to the technique in which the researcher asks the subject for information about his/her thought processes after the task is completed. Concurrent verbalization, used in this research, refers to a technique in which the subject is asked

simply to verbalize his/her thought process while working on a task (Ericsson and Simon, 1980).

Concurrent verbalization procedures have been used extensively in the study of human problem-solving, including such areas as general problem-solving behavior, physics problem-solving behavior, stock selection, pediatric cardiology, and accounting information decisions (Vitalari, 1981).

2. Validity of Concurrent Verbalization

According to Vitalari, despite the extended use of concurrent verbalization, considerable contention surrounds its use. Some researchers have questioned the validity of verbal reports. The four major issues under contention are:

- skewed verbalization of true thought process
- incompleteness
- interference with thought process
- subjective bias during analysis

The first major issue is that the subject must articulate his/her own thought process, he/she is allowed to decide how it will be verbalized. Therefore, the thought process is different than the one verbalized. The second issue, incompleteness, argues that the task of verbalizing interferes with the main task and hence, the subject is only able to verbalize a small part of the actual thought process. The third issue, interference with thought process, refers to the researcher probing the subject to explain his/her

reasoning, etc., during the experiment. The fourth issue, subjective bias, occurs if the researcher's analysis of the data is different from what is implied by the verbalizations.

Some of the ways to safeguard against the above problems include ensuring the researchers do not probe the subjects during the experiment, and having independent researchers analyze the protocols (Vitalari, 1981).

3. Evaluating Protocol Analysis Data

A wide range of methods to evaluate protocol analysis data is reported in literature, varying from a quick count of the occurrence of certain words in the protocols, to an extensive analysis of all the elements in the tasks under investigation. The method chosen to analyze the concurrent verbalizations in this research was the simple technique of searching through the protocols for unique ideas, thoughts, etc., relating to traceability issues.

D. STUDY DESIGN

1. Subjects

The subjects in our study came from a Masters program in Information Technology at the Naval Postgraduate School. The study was conducted after the students had completed the analysis, design, and implementation of an information system, based on a case study conducted in a graduate level systems analysis and design course. The case study development was based on a real-life, large-scale project and had been successfully used in similar studies (Ramesh and Dhar, 1992). The case analysis

involved a variety of data-gathering methods during the analysis phase,including informal descriptions of user needs, simulated client meetings, and actual documents from real-life situations. The major outputs developed by the participants included requirements statements, data flow diagrams, entity-relationship diagrams, database design, and implementation.

2. Case Study

The case used in this research was in the domain of customer order processing in a utility company. The problem was selected for several reasons:

- The case study had been developed after an extensive domain analysis was conducted, based on a real-life system developed by a large information systems consulting organization.
- The case study had been used successfully in several settings, including protocol analysis of group problem-solving behavior.
- The problem domain was familiar to the students, as they had personal experiences with the services provided by the system.
- Real-life data could be easily collected from a utility company and used in the analysis and design of the system when necessary (e.g., rate schedules were collected from the local utility company and used in systems design).
- The problem was sufficiently complex to cover all the basic elements of systems design.
- The problem could be partitioned so that different groups of students could be assigned projects that could be completed within a reasonable time frame.

These activities were completed during a period of over two months prior to the subjects' participation in the focus groups. Many subjects had extensive experience

in domains other than computer-based systems development, such as shipbuilding and aviation maintenance, where concepts of traceability are widely recognized.

Appendix A provides details of the case study including the various outputs produced by the subjects during the exercise.

3. Focus Groups in our Research

Six focus groups were conducted over a two-week period following the subjects' completion of their case studies. Each group consisted of approximately eight to 10 subjects and each group lasted roughly one-and-one-half hours. The focus groups were conducted in a semiformal setting--a meeting room equipped with facilities for audio/video recording. The following steps were utilized for each session:

- A short warm-up period, during which everyone, including the moderators, was introduced and the ground rules of the interview stated.
- A predisposition discussion about the traceability issues that needed to be explored, including general discussions on the various stakeholders' interest in traceability.
- A collective and comparative discussion of all topics, followed by a wrap-up of the discussion. During this segment, the participants were prompted for their summaries of what was discussed in the group meeting.

In light of the information detailed above, we felt ourselves to be on firm empirical ground in using focus groups for our research. The following are specific reasons we used this technique:

- The focus group is a valid, proven research tool in areas such as traceability, where not much is known about the topic and where generation of ideas and hypotheses for further study is desirable.

- There has been ample research on the focus group technique to give us a solid background in using it; at the same time, focus groups may be conducted informally to work well within our academic setting.

- As mentioned above, when moderators are new to a research topic, they are actually at an advantage in not reiterating the established knowledge of the field.

- The groups of students attending the Naval Postgraduate School were acquaintances who have similar socioeconomic backgrounds and levels of intelligence. At the same time, there was a small degree of variation (students from different Navy backgrounds) that Stuart and Shamdasani called for in the groups.

- Since it is preferable to have two or more analysts coding focus group results independently, this technique has proven to be suitable for a multi-person research team.

- The rolling interview technique allowed us to learn as we went along and to benefit from conducting multiple groups.

- As previously mentioned, a simple descriptive narrative, rather than technically detailed analysis of the focus groups conducted is the most appropriate method for analyzing our data.

4. Protocol Analysis in our Research

Twelve subjects who had participated in the focus group interviews volunteered for the protocol analysis portion of the data collection. This exercise started with each subject participating in a few short warm-up examples to get him/her accustomed to thinking aloud. Following these exercises, participants were handed written instructions, followed by a question-answer segment, during which clarification of their questions was provided. The exercises were conducted individually, with each subject working in a semi-private area and his/her thoughts, as they were verbalized, were tape recorded. The researchers monitored the sessions to operate the audio equipment and to prompt the subjects to verbalize their thoughts, when necessary. Each session lasted

from 30 to 75 minutes. The recordings were transcribed verbatim, and then searches were conducted throughout the transcripts for key words, phrases, concepts, or ideas that dealt with issues relating to traceability.

In view of the proven track record of protocol analysis in numerous diverse areas, it was decided this technique could also prove beneficial to our research. Some specific reasons for choosing this method include:

- Protocol Analysis was expected to provide detailed information on problem-solving with traceability information.
- A sufficient number of subjects who had prior exposure to concepts of traceability in domains other than computer based systems development and who had participated in a systems development (as a part of the case study) were readily available.
- The issues under contention, mentioned in Section C above, are minimized in our study since the safeguards discussed were implemented during the exercise.

E. SUMMARY

This chapter provided an overview of the two data- collection techniques employed, and why they were appropriate for our research. The next chapter will provide a specific analysis of the data collected.

IV. ANALYSIS OF DATA

A. INTRODUCTION

In this chapter, we discuss results from the analysis of data collected during focus groups and protocol analysis. First, we review the context in which traceability information is likely to be used during systems development; i.e. from the perspectives of key stakeholders involved in the systems development process. This is followed by a discussion of major issues that need to be addressed in the development of a model of traceability, and the mechanisms to support the capture of and reasoning with this information. Findings from relevant literature are included to elucidate the main issues.

B. STAKEHOLDERS

A number of stakeholders are involved in the systems development process, including project sponsors, project managers, analysts, designers, maintainers, and end users. The development of a model of traceability should be geared towards these various stakeholders in the systems development process. This section will address these key stakeholders and what their concerns/uses for traceability encompass.

1. Project Sponsor

A project sponsor is the individual or organization that provides funding for the system being developed. ("The project sponsor is mostly concerned about cost

overruns and a finished product.")[1] Besides assuring the sponsor that genuine requirements are met, traceability also provides a mechanism to verify that unnecessary ("wouldn't it be nice to have") features are curtailed. In so doing, the sponsor can avoid potential cost overruns, schedule slippages, etc.

2. Project Manager

The project manager is the supervisor who "plans, delegates, and controls progress to develop an acceptable system within the allotted time and budget" (Whitten, et. al., 1989). He/she is the key person held accountable for a project from start to finish, and needs to ensure that all the requirements are met. In general, he should make sure the project is finished on time, within the given budget, and that ("the project/system does what it was intended to"). A project manager uses such techniques as tracking milestones, etc., to ensure his/her responsibilities are accomplished. ("The project manager needs traceability for ... tracking milestones and ... keeping tabs on projects.") According to Brown, "Traceability provides for ease in determining phase completion and product completeness" (Brown, 1987). Traceability will also help the project manager determine when all requirements have been fully satisfied.

3. Systems Designer/Analyst

"A (software) designer often needs to trace from requirements objects to the corresponding design objects or from source code to its corresponding design or

[1]This is a direct quote from a subject participating in the protocol analysis exercise. Henceforth, all quotes from a subject, made either in a focus group or during the protocol analysis exercise, will be enclosed in parentheses and quotation marks, but no specific reference will be made.

requirements objects" (Nejmeh et al, 1989, p. 981). This use of traceability will help a systems designer determine if all requirements have been considered and specifications validated. Further, the designer needs to understand why design objects satisfy particular requirements. ("A systems designer wants traceability in order to go back to the logic.")

The systems designer is also involved following systems implementation. ("To the systems designer, traceability is extremely important as far as implementation goes ... [because he] is going to have to accommodate any design changes and [determine] the relative impact within the organization. If they don't have good traceability in the system, he [systems designer] may implement a change which ... may even cripple the syst. n.")

4. Maintainer

Maintainers are the personnel who make repairs to the system, once it has been implemented, and updates it to keep up with changing requirements. Once a change is required, a maintainer needs to be able to trace that change back to the requirements that necessitated or triggered it, and to pinpoint which parts of design/implementation are affected by the change. ("The systems maintainer wants traceability for ... tracing to a piece of code, for updating, and for changeability.")

5. End Users

Different levels of end users will employ traceability in varying degrees. On one end of the spectrum is the casual end user, "one who may use only a specific on-line

program on an occasional basis" and "may never become truly comfortable with the terminal or the program" (Whitten, et. al., 1989). An example of a casual end user is a data entry clerk. On the other end of the spectrum is the dedicated end user, "one who will spend considerable time using specific on-line programs. This user is likely to become comfortable and familiar with the terminal's operation" (ibid, 1989).

The casual end user may have little or no need for traceability. ("[Casual end users] are pretty much just concerned about using the system and don't really care or have any power over where it came from or why it is the way it is."; "I wouldn't see where they would be interested in the traceability of the design and the functionality of the system.")

Dedicated end users, however, have more applications for traceability. Some subjects noted: ("The more sophisticated end user needs traceability to manuals to see how to achieve the functionality specified in requirements documents[2] and traceability to programs, via queries, to modify them to achieve functionalities."; "For the dedicated end user, traceability is beneficial for understanding reasoning ... and for troubleshooting."). As for the end user manager, ("He needs traceability for accountability,[3] for attempting to improve on a prototype ... and to enhance documentation.").

---

[2]Traceability to documents is discussed further in a later section.

[3]Accountability is addressed in a later section.

C. MAJOR ISSUES

Our studies brought out several issues that need to be carefully examined to facilitate the development of a model of and mechanisms to capture and use traceability information. Following are some key issues we discovered, both in focus groups and protocol analysis, while evaluating the data:

1. Bidirectional Traceability

Bidirectional traceability implies both forward and backward traceability. Forward traceability is provided if each requirement specifically references a design component. In other words, forward traceability allows one actually to see where requirements materialize in the finished system.

> In the context of software design, forward traceability ... is especially important when the software product enters the operations and maintenance phase. As code and design documents are modified, it is essential to be able to ascertain the complete set of requirements that may be affected by those modifications (Thayer and Dorfman, 1990).

Backwards traceability is provided when a requirement is referenced by a design element. In the context of definition of requirements from source documents, "Backward traceability ... to previous stages of development depends upon each requirement explicitly referencing its source in previous documents" (Dorfman and Thayer, 1990). Here, bidirectional traceability indicates that a requirement derived from a former requirement has been considered, and that any new requirement can be traced back to a preceding one.

Though one of the most critical uses of traceability is ensuring that a design element satisfies a requirement, the existence of such a link may not answer the question:

are the functionalities of the design element *required by* requirements? To help answer this question, links need to be bidirectional in order to allow requirements to be traced forward from requirements to systems components, and backward, from systems components to requirements.

2. Criticality of Requirements

A useful way of identifying critical requirements is to relate them to the central "mission" of the system. Business processes or missions that *generate* requirements could be identified, and requirements evaluated with respect to such processes, to arrive at a classification. For example, traceability should address the issue of how the requirements are arrived at. This necessitates a mechanism to represent the *elaboration* and *refinement* of requirements, from the central mission or business processes that *generate* them. A good traceability scheme should recognize that all requirements are not equal in level of significance or criticality. Different levels of detail must be established in order to minimize the overhead involved in capturing and using traceability information. It may be unnecessary or even undesirable, considering the overhead involved in maintaining traceability, to maintain linkages between every requirement and every output created during the systems design process related to it. Costs must be justified by the benefits. It is essential to identify critical requirements and maintain traceability from those requirements to the various systems components.

The need to relate mission criticality to a traceability scheme was considered important by many subjects in the focus groups: ("We just have to realize that it

[traceability] is not necessary for mundane decisions."; "Traceability is great for the critical stuff.").

3. Design Rationale

Another important component of traceability is design rationale information. On the need for design rationale MacLean states: "To understand why a system design is the way it is, we also need to understand how it could be different and why the choices which were made are appropriate" (MacLean, et al, 1989).

Traceability linkages to represent rationale would capture the *why* or *reason for* design decisions. Design rationale allows for reasoning about a system's characteristics in the process of understanding and changing it. Design rationale is an important issue in change management, as it can facilitate change while not necessarily providing the mechanism for doing so. Tracking relationships among design objects, and understanding how and which of those objects is affected by change, is vital in the maintenance of the system.

The focus group participants were keenly aware of the need for design rationale as a component of traceability: ("The systems designer needs traceability in order to examine the logic behind the system."; "Traceability could be very useful for justifying why you did something the way you did it."; "Traceability would be good for determining what input and output are required."; "We have some artificiality built into the system--you can say this is how it's supposed to be, but is it really? You may need traceability to help you adjust requirements.").

4. Project Tracking and Management

Requirements traceability can be used very productively in project management and tracking. During the systems definition and subsequent phases, traceability is essential to ensure that all systems requirements have been met. Establishing all life cycle phases as complete can go a long way toward guaranteeing the ease of the verification and validation process.

The project manager can use links such as *status, completion date,* and *authorization* between various components of the system for scheduling, continuity, and security. Such information is indispensable in integrating project management into the systems development operation, and the efficient completion of project management tasks.

Focus group participants were very interested in project tracking and management possibilities using traceability. ("Without traceability, if you've lost a linkage you spend much valuable time tracing back to the original requirement."; "If you don't write down your thought processes and assumptions, and most people don't, you can't remember what you've been doing unless you have traceability."; "Humans don't go back to the requirements enough."; "Traceability should be extremely helpful with tracking costs."; "The project manager needs traceability for tracking milestones."; "Traceability would be great for the project manager's security concerns.")

5. Accountability

A major use of traceability is to provide accountability. Using traceability legitimately to communicate with the original designer of a system component, or to understand the capability of a system, is an example of such potential use. However,

caution must be used when employing traceability information to enforce accountability. The use of accountability information as a means for performance appraisal may be inappropriate. A parallel could be drawn to the use of information gathered during structured walkthroughs in systems development which should be strictly used for understanding and improving the current system and not for performance evaluation.

Some accountability information that could be captured using traceability linkages include: design elements *designed by*, *validated by*, and *modified by* development personnel. The availability of such information will be indispensable in maintaining and revising a system.

The focus group subjects perceived an urgent need for the accountability element tempered by constraints on its usage: ("Traceability needs to be something that humans can work with, not just a whip held over people."; "Traceability should not be used to threaten people with."; "Accountability needs to be supplemented with good communication."). They were also mindful of the future: ("Accountability implies affordability--we're going to have less resources available in the future."; "I'm sure that I'm going to want to look back in the future and ask myself who made certain decisions or where decisions came from.").

6. Humanware

Humanware form a critical component of any large-scale system. It just as important to capture how requirements relate to humanware as with other components of the system. This may involve tracing the responsibility for a requirement to a human.

A comprehensive mechanism for traceability should link the humanware component of a system to the other components. Examples of such linkages include systems functionalities *performed by* humans. This information is necessary to ensure that the allocation of requirements is complete and correct. Focus group participants touched on the concept of humanware: ("We need traceability for human manageability.").

7. Documents/Manuals

Document traceability determines the existence of relationships between two document segments; it means that a particular document is in accord with a previous document, with which it has some type of relationship. Document traceability also ensures that all components in one document have a related component in another document.

> Consistency and completeness constraints apply within a document and across documents. Within a requirement specification, a requirement description may define inputs and outputs which relate to other requirements in the specification. Inconsistent references and incomplete specifications may occur and can be checked... (Horowitz and Williamson, 1986).

Traceability linkages to documents include *interpreted by*, *defined by*, and *consistent with*. Such linkages specify how to obtain a required performance from a systems component. Our focus group subjects had considerable insight into some of the document traceability implications: ("Stakeholders are interested in having traceability to be able to write quality manuals and data dictionaries."; "Traceability is good in that it de-emphasizes [unnecessary duplication in] documentation."; "If traceability is good, another contractor should be able to do documentation."; "Traceability

is not a requirement of documentation, but it is highly desirable for documentation purposes.").

8. Dependencies

Since complex systems are composed of interdependent components, such *dependencies* should be represented and maintained. Often the inter-component dependencies are not well understood and documented.

Systems design is a complex activity involving interdependent decisions. In the absence of mechanisms to record such dependencies, over time and with changing development teams, this information will be lost. Such dependencies may span different systems components. A decision about software may be dependent on an earlier decision about hardware. As the system evolves over its life cycle, the hardware decision may be altered, leading to inconsistencies with the software that was based on the earlier hardware decision. Unless the dependencies are captured and maintained, such issues may go undetected, leading to severe system integration problems.

Another form of dependency is the fact that there may be several components needed to satisfy a requirement. As the system evolves over its development life cycle, it is desirable to identify design or implementation elements that "partially satisfy" a given requirement. For instance, a hardware/software combination is often necessary to satisfy a given requirement. When either the hardware or software component is developed, traceability information should reflect the fact that the partially satisfied requirements are fully satisfied by performance of necessary actions.

It is possible to identify a combination of design elements that *satisfy* a requirement or are *generated by* it. An example of such a traceability scheme is the use of AND-OR graphs to represent traceability linkages. Such AND-OR graphs can be used to model a task in terms of a series of goals and subgoals. If requirements are treated as goals to be satisfied, the successive refinements can be treated as subgoals to be satisfied. The goals which can be satisfied only when all of their immediate subgoals, are satisfied are represented by AND nodes. When goals can be satisfied by any of the subgoals, they are represented by OR nodes. Liu and Horowitz (Liu and Horowitz, 1989) model the Work Breakdown Structure (WBS) of a software project as an AND-OR graph. This concept can be used in maintaining traceability linkages between various levels of outputs, when a logical combination of lower level outputs satisfies a higher level goal or requirement. An AND-OR graph is depicted in Figure 1.

Yet another form of dependency can be summed up: ("The data base design has a *transitive dependency*."). This dependency is identified when the ("data base design requires the data flow diagram [which in turn] depends on the requirements. Therefore, requirements determine database design.").

9. Horizontal and Vertical Traceability

Vertical traceability refers to the "association of software (system) life cycle (SLC) objects of different types (typically created in different SLC processes). An example of a vertical traceability relationship would be between requirements statement and design statement" (Nejmeh et al, 1989, p. 981). ("Vertical traceability is easy because there's a 'rule' ... you explode a process and either you have to or you don't.").

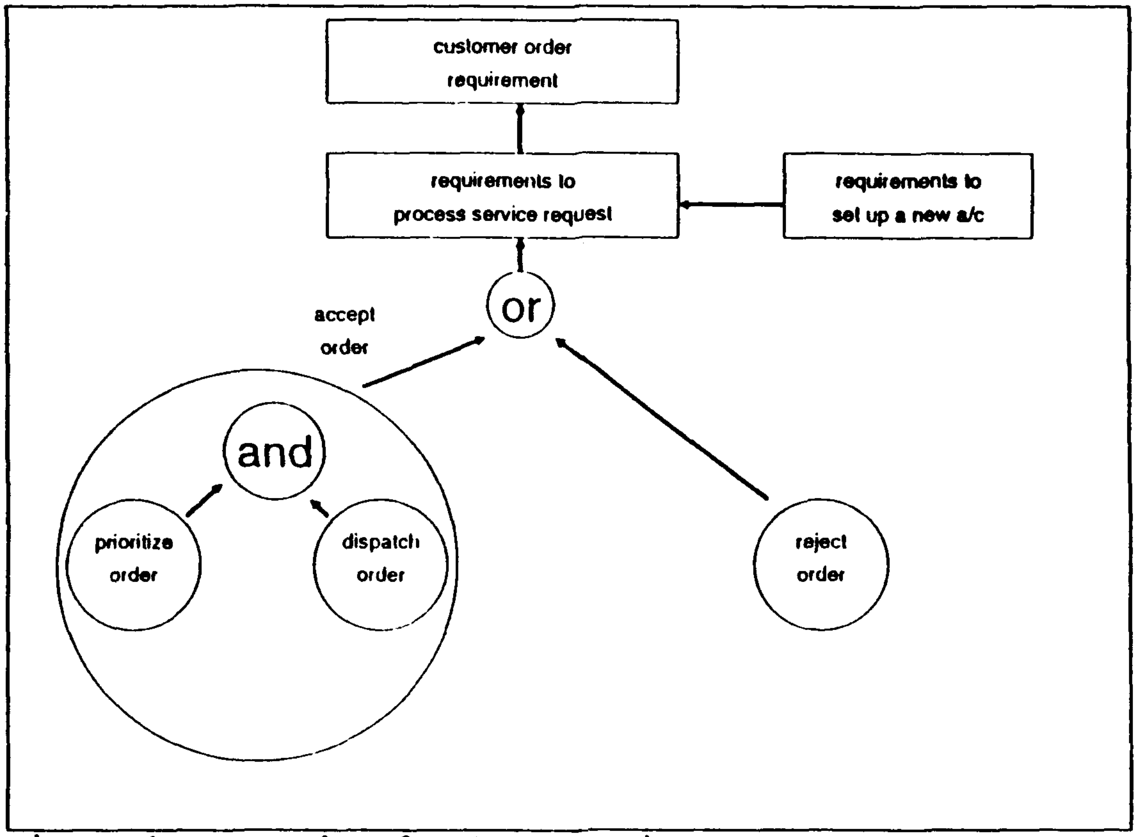

Figure 1. Example of AND-OR Graph

Horizontal traceability refers to the "association of SLC objects of the same type (typically created in the same SLC process). An example of this type of traceability includes parent/child relationships among decomposed data flow diagrams and the 'derived from' relationship among requirement statements" (Nejmeh et al, 1989, p. 981). ("When you're moving horizontal is when you're analyzing what process is inside what process."). Horizontal traceability equates to a ("subprocess transferring data to another subprocess like primitive levels have to talk to each other, etc.").

Horizontal traceability also refers to relationships between different views of the same level of (design) representation. For example, the relationships between the

behavioral view and functional view of the system [Hoang, 91] could provide horizontal traceability.

10. Automated Support for Traceability

Automated support for traceability can be extremely valuable when systems are large and/or complex. "When performed manually, the tasks are time-consuming and error-prone; moreover, users' abilities to analyze traceability data are limited by the sheer volume of data ..." (Nejmeh et al, 1989, p. 981). In such circumstances, "an automated software tool is an imperative, as the measuring process can become extremely onerous" (Shepperd and Ince, 1990). As stated by Thayer and Dorfman, "There have been many cases where it appeared, at the outset, that it would be an easy task to keep track of it [manually], but when the system design is complete, and the customer is trying to understand whether all the test data really satisfies the original requirements they wrote, the automated traceability would be 'worth its weight in gold'" (Thayer and Dorfman, 1990).

The degree of automated support can vary widely, depending on the level of sophistication warranted/desired. "The simplest [form] is a list that is tabulated by the ID of the requirement" (ibid, 1990). This list can be changed, as needed, to support the iteration process. The use of a flexible database program and other more intricate aids can be utilized for more complex automated support.

D. SUMMARY

The issues reviewed above suggest there are many aspects of traceability which need to be considered when contemplating a traceability model for real-time, complex systems. This chapter specifically indicates that different stakeholders will have different uses for traceability, and in varying degrees. The next chapter provides a model of design rationale as an example of a complex traceability relationship to illustrate the concepts discussed here.

V. DESIGN RATIONALE AS AN EXAMPLE OF TRACEABILITY

A. INTRODUCTION

In this chapter, we discuss a model for representing and reasoning with design rationale as an example of a complex traceability scheme to illustrate and highlight some of the major issues discussed in the previous chapter.

A conceptual model and mechanisms for the representation of and reasoning with process knowledge (i.e., design rationale) have been developed in earlier research as a part of the REMAP (Representation and Maintenance of Process Knowledge) project. The model and the mechanisms provided by REMAP for representing and reasoning with traceability information to support various stakeholders is discussed in detail elsewhere (Ramesh and Dhar, 1992). This design rationale model can be viewed as an instance of a traceability link between a requirement and a design element. The term "design element" denotes any part of the system design or implementation (i.e., data flow diagrams, specifications, pieces of hardware, humanware etc.). In this chapter, we discuss how such a model and reasoning mechanisms can be used in the context of the issues discussed in the previous chapter.

B. ISSUES IN CAPTURE AND USE OF RATIONALE

1. Support for various stakeholders

There are a variety of stakeholders involved in large software projects, each having a different set of goals and priorities. For each of the stakeholders, some useful support can be provided by recording in some structured manner, the history of a design in the form of (design) rationale.

2. Partially Satisfied Requirements

The process of satisfying requirements may generate several issues that need to be resolved. Resolution of issues lead to one or more design components. Partially satisfied requirements may be identified with unresolved issues that relate to that requirement using structures like the AND-OR graphs in REMAP. A similar structure can be used in linking design artifacts to requirements through design decisions.

3. Criticality of Requirements

Our model captures the elaboration and refinement of requirements. Critical 'mission statements' or core 'business process' objectives can be the origin of such an elaboration and refinement. During this process, the criticality or importance of requirements can be ascertained and monitored. The REMAP model can represent this information as an attribute of the links between mission statements/business processes and requirements or as attributes of requirements themselves. Then, the critical requirements can be monitored to ascertain whether all the issues related to them are resolved in a timely manner.

4. Qualitative and quantitative reasoning

The strength or other characteristics of relationships can be either qualitative or quantitative. In REMAP, the contents of the primitives can be informal information (such as text). But the model has well defined semantics of relationships among its primitives, facilitating reasoning with this structure. For instance, the assumptions in a design situation can be given different degrees of belief (or validity), and these beliefs can be automatically propagated to beliefs in arguments, positions and so on. Further, the strengths can be either qualitative or quantitative.

5. Change Management

In REMAP, changes to design rationale will automatically trigger changes in the belief status (or validity) of design solutions thereby suggesting redesign (Ramesh and Dhar, 1992). Since various components of the process knowledge that lead to the design solution are tightly related, changes to the constraint set resulting out of changed assumptions, decisions or requirements will initiate the synthesis of a new design solution and provide rich information to estimate the effort involved in redesign.

6. Project Management

REMAP provides facilities for representing and reasoning with temporal information which can be useful for project management. For instance, a validity time can be assigned to issues which could be interpreted as the time frame during which that issue must be resolved. Then, this information can be used for generating reminders to the designers or managers to focus their attention on issues that may have to be resolved within a time

frame or used in rank ordering issues. Project planning and control can be facilitated by integrity constraints on its primitives. An example of such a constraint could state that no requirement can be elaborated or refined until all requirements with higher priority or earlier validity time are considered.

7. Accountability

The REMAP environment facilitates the automatic capture or the representation of accountability information associated with design rationale.

8. Links to all system components

The REMAP model can be used to capture relationships between requirements and all system components, including humanware, hardware, software etc.

9. Automated Support

REMAP provides automated support for different stakeholders including interactive querying and updating of the design rationale knowledge base, a client-server architecture for multi-user support, a textual as well as hypertext-like user interface to the knowledge base and a reason maintenance system for maintaining and reasoning with design rationale.

10. Derived Links

REMAP provides facilities for inferring knowledge based on deductive rules and facilitates the derivation of implicit links between requirements and design artifacts. For instance, a rule could state that if a design element is created by a decision, and the

decision resolves an issue and the issue was generated by a requirement, then the design element traces to the requirement.

C. SUMMARY

Design rationale information supports a variety of stakeholders. A semantic model of design rationale such as the REMAP model illustrated here is essential for providing such support, facilitating reasoning with such knowledge.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

In this chapter, based on the findings discussed in Chapter IV, an initial model of traceability is presented. Further, recommendations on methodologies to be used in a comprehensive study on traceability are presented. Specifically, the appropriateness of the two techniques used in this research are discussed.

B. INITIAL MODEL

The findings from the preliminary study suggest that a comprehensive model of traceability needs to be developed. Our approach to developing a model is to understand the traceability needs of various stakeholders in the systems development process. In order to fully support the stakeholders needs, our research suggests that a comprehensive model of traceability should capture semantic information (as does the REMAP model for design rationale) to allow for advanced reasoning with the traceability data. An initial attempt at a traceability model is shown in Figure 2 which demonstrates linkage, linkage types and ways to combine the links.

In this figure, various links are shown between different stages of the development process (e.g, requirements and design). Recursive relationships are shown to denote both vertical (e.g., between high level and low level design) and horizontal (e.g., between different representations or views at the same level of design) linkages within a

development stage. Each link consists of at least one traceability type. At present our model consist of five types of links as denoted in Figure 2. These types were derived from the analysis of data explained in Chapter IV. An example follows: Requirement 1.3 from document A is linked to function X (represented by dataflow diagram bubble 2.4). The links is of type satisfied which denotes that function X satisfies requirement 1.3. In some cases links may include more than one type. For example, the above link may be augmented by accountability information (e.g. the satisfied relationship was determined by Mr. Smith on Sept 10, 1992).

In order to better support reasoning of the traceability information, the model allows for several methods of combining the traceability links. The four methods for combining, shown in Figure 2, were derived based on the data collected. The need for a weighing scheme was noted by the discussion on criticality and the other three are described in detail in Chapter IV). An example of an and/or scheme was also presented in Figure 1.

The authors believe that by using various types of links and methods for combining them, like the ones presented in this model, one can adequately capture the traceability information and provide reasoning to satisfy the stakeholders previously mentioned needs.

C. METHODOLOGIES FOR FUTURE RESEARCH

In this research, two very powerful techniques were employed for data collection. Though the techniques are widely used in other disciplines, they are unconventional as

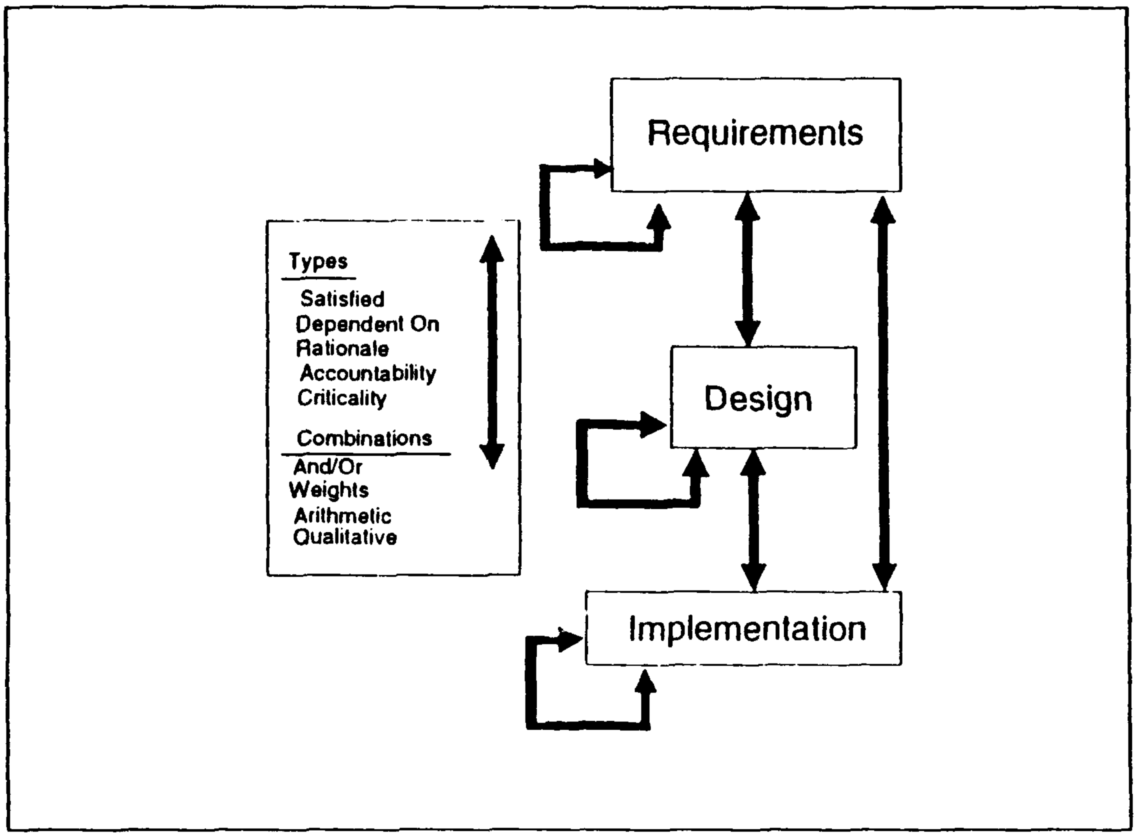

Figure 2. Initial Traceability Model

empirical research tools in the domain of systems development. Comparison of the two primary data collection strategies provides some interesting insights into the research methodology appropriate for future work. Focus groups provided surprisingly interesting results. In this exploratory data collection method, the researcher's biases do not constrain the participants. It should be noted that the moderators of the focus groups were non-experts in the domain, and therefore, the potential for biases is minimal. In our study, many participants related concepts of requirements traceability to their experiences in ship-building and aircraft maintenance which employ similar concepts. Focus groups conducted participants with real-life systems development experience is likely to provide

very valuable information, even if the participants are not very familiar with current traceability tools and techniques, provided they have sufficient interest in the concept. As the participants are not restricted by the researcher's ideas and predispositions, this methodology will often provide new perspectives and approaches to the problem being explored.

During the 1992 Complex Systems Engineering Synthesis and Assessment Technology Workshop held at the Naval Surface Warfare Center Dahlgren Division, a break-out session on Traceability was held. Participants included Systems designers as well as CASE tool developers. The meeting had some of the characteristics of a focus group, though held in an informal setting. Several of the major issues that were identified in our study were raised by this informal focus groups, providing an informal validation of our findings. Further, as the group members had the characteristics similar to those of potential subjects in our comprehensive study, it is believed that focus groups as a methodology for data collection will be very valuable.

Protocol analysis, on the other hand, is likely to provide very specific data on problem solving behavior. Very useful results can be obtained if the behavior is studied in a real-life problem solving situation. This requirement severely constrains the use of protocol analysis in our future work. First, current methods of capturing and reasoning with traceability are inadequate to provide us an appropriate real-life problem solving situation. Second, the protocol analysis method is extremely expensive in terms of demands on the subjects and the researchers. Therefore, the use of this methodology

should be restricted to a very small number of subjects in a relatively well-defined area of problems solving (e.g., traceability for accountability).

D. CONCLUSIONS

Our research provides very valuable insights into the development of a comprehensive model of traceability. Continued research needs to be done to refine and validate the model. The link types need to be further defined and the use of different traceability types in system development activities needs to be explored. The methods for combining links needs to be examined further. Further, automated methods for capture and analysis of links that involve various methods of combining them would be very useful. Over the next year the research intends to finalize the model and prioritize the types and combinations in terms of their importance in supporting various stakeholders. In future years each link type and combination method will be further investigated to enhance the model.

Appendix A. CASE STUDY DESCRIPTION

The project provided an opportunity to learn about systems analysis and design by performing the analysis and design of a system based on a case study. The case study had been developed after a detailed domain analysis. The domain analysis used data from a real-life large scale systems development effort.

REQUIREMENTS

The participants were required to produce several outputs at various stages.

PHASE 1: PROJECT PROPOSAL

In this phase, the participants identified the application to be studied and provide the motivation for their project.

A typical project proposal including the following:

1. Name of the Project
2. Brief description of the project
   - 2.1 Background
   - 2.2 Management
   - 2.3 Data Processing at the Organization
   - 2.4 Concerns over Information Systems issues that motivate the project

3. Brief background of each team member and proposed division of tasks for the project (with justification).

4. Preliminary investigation to determine information requirements

    4.1 Interview reports (of key people involved)

    4.2 Summary of findings (This is typically a verbal description of the key elements found in all the interview conducted and reported in 1.1)

    4.3 Formalized Requirements Statements based on preliminary investigation (These were to be considered similar to those produced during Govt. systems development efforts)

<u>PHASE 2: SYSTEMS ANALYSIS</u>

Typically, in this phase, data flow diagrams, entity-relationship diagrams and data dictionaries are developed for the system. The basic tasks performed by the participants included:

1. Data Flow Analysis

    This analysis consists of developing data flow diagrams, which describe the processes and the data dictionary, which defines system elements. Various CASE tools were used to develop the components.

2. Entity-Relationship Model

    This analysis involves the use of the entity-relationship model to document the systems's data independently of how the data will be used.

PHASE 3: SYSTEMS DESIGN

The participants designed and implemented the systems using a relational database system. The database design phase included development of the logical schema and normalization. The project also included the development of a man-machine interface of the system being designed. Application generators were used to create input and output layouts.

CASE STUDY PROJECT TOPICS

The participants were chose one of the four following subsystems of a customer order processing system for a utility company.

Subsystem 1: **Telephone Answer Center System**

This subsystem deals with the operations of the telephone answer center. The primary users of the system are likely to be the telephone answer center operators and supervisors.

Subsystem 2: **Field Station System**

This subsystem deals with the operations of the field offices for handling customer requests (except for appliance repair and maintenance). The primary users of this system are likely to be the field supervisors and field technicians.

Subsystem 3: **Appliances Repair and Maintenances System**

This subsystem is a specialized field station system for handling appliance repairs and maintenance. The primary users of this system are field supervisors (appliance) and field technicians.

Subsystem 4: **Billing System**

This subsystem deals with periodic computation of bills to be provided to customers for the services provided by the utility company. The primary user of this system is the billing department.

LIST OF REFERENCES

Agusa, K., Ohnishi, A., and Ohno, Y., "A Verification Method for Formal Requirements Description," *Journal of Information Processing*, v. 7, 1984.

Brown, B. J., "Assurance of Software Quality," SEI Curriculum Module SEI-CM-7-1.1 (Preliminary), Carnegie Mellon University, Software Engineering Institute, July 1987.

Dorfman, M., and Thayer, R. H., *Standards, Guidelines, and Examples on System and Software Requirements Engineering*, IEEE Computer Society Press, 1990.

Edwards, M. and Howell, S., "A Methodology for Systems Requirements Specification and Traceability for Large Real-time Complex Systems," , Technical Report, Naval Surface Warfare Center,
August 1992.

Ericsson, K. A., and Simon, H. A., "Sources of Evidence on Cognition: A Historical Overview," in Merluzzi, T. V., and Glass, C. R., eds., *Cognitive Assessment*, New York Guilford Press, 1980.

Greenspan, S. J., and McGowan, C. L., "Structuring Software Development for Reliability," *Microelectronics and Reliability*, v. 17, 1978.

Horowitz, E., and Williamson, R. C., "SODOS: A Software Documentation Support Environment—Its Use," *IEEE Transactions on Software Engineering*, v. SE-12, November 1986.

Krueger, R. A., *Focus Groups: A Practical Guide for Applied Research*, Sage Publications, Inc., 1988.

Liu, L., and Horowitz, E., "A Formal Model for Software Project Management," IEEE Transactions on Software Engineering, v. 15, October 1989.

MacLean, A., Young, R. M., and Moran, T. P., "Design Rationale: The Argument Behind the Artifact," in *Conference Proceedings of Human Factors in Computing Systems*, Austin, Texas, May 1989.

Morgan, D. L., *Focus Groups as Qualitative Research*, Sage Publications, Inc., 1988.

Nejmeh, B. A., Dickey, T. E., and Wartik, S. P., "Traceability Technology at the Software Productivity Consortium," in Ritter, G. X., ed., *Information Processing '89*, Elsevier Science Publishers B. V., 1989.

Ramesh, B., and Dhar, V., "Supporting Systems Development Using Knowledge Captured During Requirements Engineering, submitted to *IEEE Transactions on Software Engineering*, June 1992.

Ramesh, B., and Edwards, M., "Issues in the Development of a Requirements Traceability Model," in proceedings of *IEEE International Symposium on Requirements Engineering*, San Diego, California, January 1993.

Schneidewind, N. F., "Software Maintenance: Improvement Through Better Development Standards and Documentation," Naval Postgraduate School, Monterey, California, February 1982.

Shepperd, M., and Ince, D., *Multi-dimensional Modelling and Measurement of Software Designs*, 1990 ACM Computer Science Conference, Washington, DC, February 1990.

Stewart, D. W., and Shamdasani, P. N., *Focus Groups: Theory and Practice*, Sage Publications, Inc., 1990.

Templeton, J. F., *Focus Groups: A Guide for Marketing & Advertising Professionals*, Probus Publishing Company, 1987.

Thayer, R. H., and Dorfman, M., *System and Software Requirements Engineering*, IEEE Computer Society Press, 1990.

Vitalari, N. P., *An Investigation of the Problem Solving Behavior of Systems Analysts*, Ph.D. Dissertation, Graduate School of the University of Minnesota, Minneapolis, Minnesota, pp. 82-84, 87, 89, June 1981.

Vitalari, N. P., and Dickson, G. W., "Problem Solving for Effective Systems Analysis: An Experimental Exploration," *Communications of the ACM*, v. 26, November 1983.

Whitten, J. L., Bentley, L. D., and Barlow, V. M., *Systems Analysis & Design Methods*, 2nd ed., Richard D. Irwin, Inc., 1989.

BIBLIOGRAPHY

Baldo, J., "Reuse in Practice Workshop Summary," Institute for Defense Analysis, April 1990.

Fischer, A., "CASE Tool Gets Friendly Enhancements," *Electronic Engineering Times*, 12 September 1988.

Hadfield, S. M., *Interactive and Automated Software Development*, Master's thesis, Air Force Institute of
Technology, December 1982.

Horowitz, E., and Williamson, R. C., "SODOS: A Software Documentation Support Environment—Its Definition," *IEEE Transactions on Software Engineering*, v. 12, August 1986.

Keuffel, W., "Extra Time Saves Extra Money," *Computer Language*, December 1990.

Macmillan, J., and Vosburgh, J. R., "Software Quality Indicators," Scientific Systems, Inc., September 1986.

"Military Fine-tuning Urges Development Standards," *Software News*, November 1986.

Murine, G. E., "Secure Software's Impact on Reliability," *Computers and Security*, v. 5, 1986.

O'Brien, D. H., "Software Quality Starts with the Customer," *Quality*, June 1991.

Wrigley, C. D., and Dexter, A. S., "A Model for Measuring Information System Size," *MIS Quarterly*, June 1991.

Wuebker, F. E. "The Impact of Nebula, MCF, and ADA on Real-Time Embedded Computer Systems," RCA Government Systems Div., November 1982.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5002

Department of Administrative Sciences
Attn: Code AS/RA
(Prof. Balasubramaniam Ramesh) 10
Naval Postgraduate School
Monterey, California 93943-5000

Library of Congress 2
Attn: Gift and Exchange Division
Washington, DC 20540

Naval Air Development Center
Attn: Code 7033
(Dr. C. Schmiedekamp) 1
(P. Zombori) 1
Warminister, PA 18974-5000

Office of Naval Technology
Attn: Code 227
(Dr. Elizabeth Wald) 1
800 N. Quincy Street
Arlington, VA 22217-5000

Center for Naval Analyses 2
4401 Fort Avenue
P.O. Box 16268
Alexandria, VA 22302-0268

Naval Reserach Laboratory
Attn: Code 5534

(Connie Heitmeyer)	2
(Bruce Labaw)	1
Washington, DC 20375	
United States Army	
CECOM, C2NVEO	
Attn: AMSEL/RD/VNT/TST	
(H. Nguyen)	1
Fort Belvoir, VA 22060	
Advanced Technology & Research Corp.	
Attn: Adrien J. Meskin	1
George Stathopolos	1
14900 Sweitzer Lane	
Laurel, MD 20707	
General Electric-Aerospace	
Attn: Steven Sietz	1
Mail Stop 127-333	
199 [illegible]ton Landing Road	
Moorestown, NJ 08057-3075	
Bell Laboratories	
Attn: Suzanne Edwards	1
Room 3B-326	
101 Crawfords Corner Road	
Holmdel, NJ 07733	
Naval Surface Warfare Center Dahlgren Division	
Attn:	
U302 (P. Hwang)	20
U33 (S. Howell)	10
U33 (M. Edwards)	10
D4	1
E231	1
E232	1
E342 (GIDEP)	1
F01	1
F31 (W. Laposata)	1
G07 (F. Moore)	1
G42 (T. Dumoulin)	1

G42 (A. Farsaie)	1
G42 (J. Moscar)	1
G42 (E. Ogata)	1
G70 (D. Dorsey)	1
G72 (H. Parks)	1
H32 (J. Miller)	1
K02	1
K10 (J. Sloop)	1
K12 (J. O'Toole)	1
K13 (D. Parks)	1
K14 (D. Clark)	1
K41 (L. Gross)	1
K51 (J. Smith)	1
K52 (G. Brooks)	1
K52 (W. Farr)	1
K52 (H. Huber)	1
N15 (M. Wilson)	1
N33 (J. Sizemore)	1
N35 (M. Masters)	1
N35 (F. Riedl)	1
R44 (E. Cohen)	1
R44 (H. Szu)	1
U	
U02	1
U042	1
U10	1
U20	1
U23 (W. Dence)	1
U23 (R. Fitzgerald)	1
U23 (J. Horner)	1
U23 (M. Richards)	1
U23 (P. Winters)	1
U25	1
U25 (D. Bergstein)	1
U25 (E. Hein)	1
U30	1
U33	1
U33 (D. Choi)	1
U33 (K. Murphy)	1
U33 (N. Hoang)	1
U33 (M. Jenkins)	1
U33 (T. Moore)	1
U33 (C. Nguyen)	1

U33 (T. Park)	1
U33 (H. Roth)	1
U33 (M. Trinh)	1
U33 (P. Wellenberger)	1
U40	1

10901 New Hampshire Avenue
Silver Spring, MD 20903-5000